PivotalTracker

Agile Doesn't Scale—It Multiplies

By Jeremy Jarrell

Agile Doesn't Scale—It Multiplies

Agile practices have seen tremendous growth in recent years. More teams than ever have embraced a lightweight way of working and have found themselves more responsive to the needs of both their customers and their broader market. Teams adopting an agile way of working can be found in nearly every industry and every type of organization; however, big enterprises have seen one of the largest shares of growth.

While there are many reasons for an enterprise to consider a move to agile practices, the most common reason is a need to compete with smaller, more nimble competitors. These competitors are often able to respond to customer feedback in a more timely manner than their bigger and more established counterparts, which has forced many large enterprises to consider alternative approaches to delivering value to their customers.

But for many organizations who have seen success embracing agile practices, these early advances have only begged the question: *How do we scale this success to our entire organization?* In fact, the search for the answer to that question has become so popular that it has spawned a multitude of new methodologies, certifications, and even conferences all dedicated to the single question: *How do you scale your success with agile?*

Before you consider how to scale agile in your organization, you must first consider what the specific goals are that you wish to achieve with scaling.

Not so fast

Before you consider how to scale agile in your organization, you must first consider what the specific goals are that you wish to achieve with scaling.

When an organization begins to consider scaling their agile practices, they are often interested in scaling the early benefits they've experienced with their first agile team. For example, if that first team was able to deliver a successful product more quickly than had previously been possible, then increasing that organization's investment in the same team should result in that team producing even more value next time. Right?

While understandable, unfortunately this approach is often mistaken. One reason is that it demonstrates a subtle misunderstanding of the benefits that adopting an agile process actually brought to the organization.

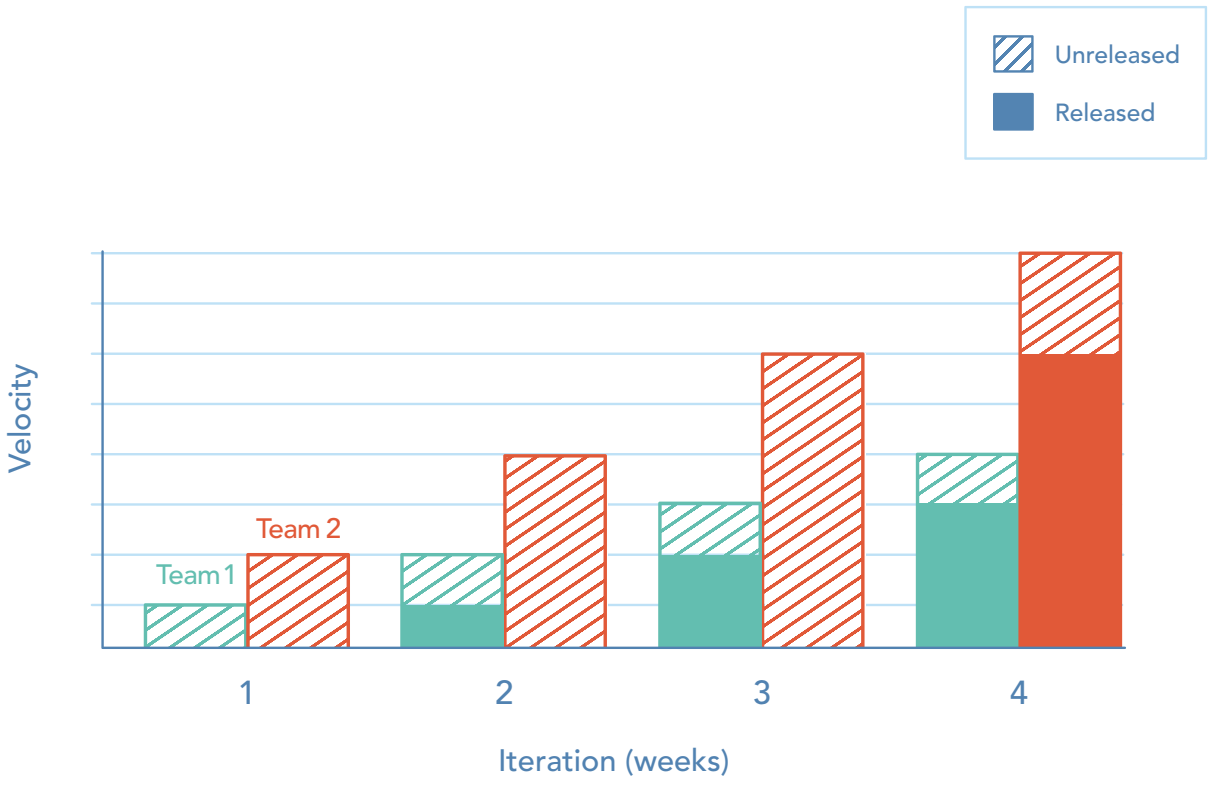
Faster vs. more frequent

Despite popular misconceptions, agile teams do not go faster than non-agile teams. In fact, given that many teams beginning an agile practice must adopt a new way of working, it's not unusual for those teams to actually slow down while adjusting to this new way of life.

So why is it that many agile teams appear to produce work more quickly than their non-agile counterparts? The answer is that successful agile teams reduce their time to market by shipping more frequently, rather than simply working faster.

Teams who ship more frequently (like Team 1 in the graphic below) experience a faster time to market while working at the same—or even reduced—rate as teams who only ship infrequently (e.g., Team 2).

This subtle difference between working faster and shipping more frequently is critical for your organization to understand if it is to have any success in scaling its agile efforts. Otherwise, you risk amplifying the wrong aspects of your team by attempting to capitalize on this misunderstood success.



The fallacy of *bigger*

So what happens when an organization misunderstands the true roots of their agile success? If they believe that their teams are actually working faster—and thus producing more work—as a result of adopting agile practices, then that organization may be tempted to try and build on that success by simply increasing the size of that team.

Many of the qualities that make agile teams successful can also be notoriously tough to scale.

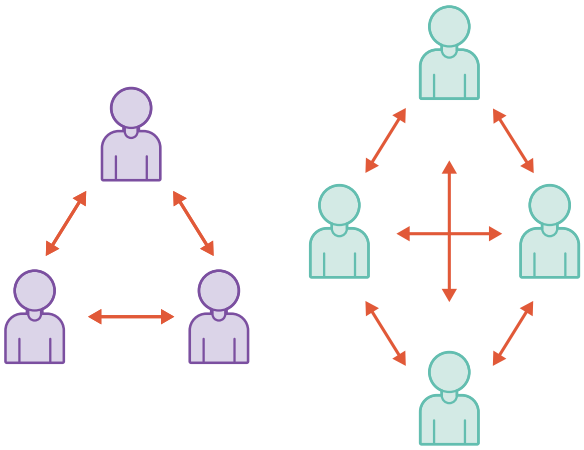
While most organizations instinctively understand that doubling the size of a successful team will not consequently double the output of that same team, few can resist the temptation to add a few new members to a team whenever a convenient opportunity arises. While agile teams are not immutable and regular team changes are a way of life for many organizations, the danger lies in believing that any increase in team size will also automatically result in an increase in output.

Successful teams can have their ranks increased and an organization can reap corresponding gains in productivity as a result of these increases. But these increases are best made in a small and incremental manner and even then, the resulting increases in productivity still may not arrive for some time.

Why scaling can be so hard

Many of the qualities that make agile teams successful can also be notoriously tough to scale. For example, agile methodologies place an emphasis on interpersonal communication over written documentation. While few would doubt that interpersonal communication conveys a depth of information far greater than written communication, it's also true that conversations become more difficult when more people are added to them.

As teams increase in size, keeping all members of that team up-to-date about the latest news relevant to their teams becomes increasingly difficult, as the number of communication pathways increases exponentially along with the size of that team.



In addition, the chances for misinterpretation and confusion also begin to increase dramatically along with team size.

But while scaling an agile team's strengths can be difficult, amplifying their problems happens almost naturally.

But while scaling an agile team's strengths can be difficult, amplifying their problems happens almost naturally.

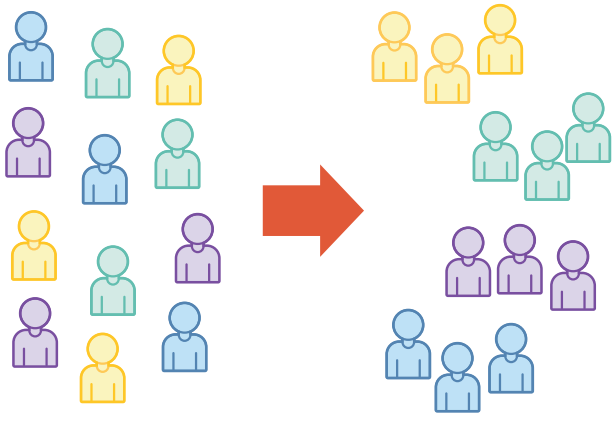
Many teams find that when they first find success with agile methodologies, this new-found responsiveness to customer needs can lead to a high degree of churn in their product plans. If it's not brought under control quickly, this churn only increases as the size of the team grows. As a result, the team may soon find itself thrashing as it moves quickly from one product idea to the next, rarely stopping to give each idea the attention necessary to truly deliver the desired end result to the customer.

But some problems plaguing a team aren't due only to that team's success with agile; some problems may lurk regardless of a team's chosen approach. For example, you may find that your team has personality conflicts that have started to emerge, or rifts that are starting to grow between different factions. In either case, if you fail to address these problems before investing in increasing the size of your team, you're likely to find that scaling your team will only serve to amplify these problems, not alleviate them.

An alternative approach to scaling

But what if there was a way to capitalize on the benefits your organization has gained by adopting agile practices without incurring the problems that can arise from increasing your team size? It turns out the best way to increase your organization's investment in agile is not to scale it, but to multiply it.

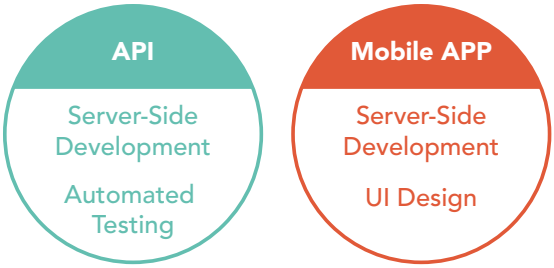
What if instead of increasing the size of a single agile team, you simply created more agile teams? Creating multiple, smaller teams can often be a more successful path to scaling agile since small team sizes naturally alleviate the exponential explosion in communication pathways discussed earlier.



Creating a cross-functional team

However, successful agile teams are more than just small—they are also cross-functional. Cross-functional teams have all of the skills necessary to bring an idea from fruition to reality within the context of the work they are undertaking. While this may sound ambitious, it's actually easier than you may first think.

For one reason, for a cross-functional team to be successful, they only need the specific skills that are necessary to deliver their piece of the overall product. For example, a team developing an API that will be delivered as a microservice does not need UI design skills or even front-end development skills to be successful. Instead, the cross-functional skills that this team needs to be successful may be limited to those like strong server-side development, RESTful API design, and robust automated testing.



On the other hand, a team developing a mobile app that will consume this API may only require native mobile app development and UI design skills to be successful. In either case, each team will have its own specific needs that will dictate which skills are necessary for them to truly be considered cross-functional.

Also remember that only the team itself needs to be cross-functional—not every person on the team. In the example of our API team, this means that every team member does not need to possess strong server-side development skills and good automated testing skills. These skills only need to be present across the team as a whole.

However, while not every individual on the team needs to possess the full breadth of skills necessary to successfully deliver the product to market, each individual does need to be adept at collaborating with other team members who possess different skill sets. This is because one skill can only amplify the other skills if the holders of those skills can successfully work together.

Finally, remember that the skills the team will need in order to remain cross-functional are likely to evolve over time. For example, returning to our API team, as the team nears their first release, they may also need the skills necessary to successfully deploy containerized microservices to their production environment. As a result, this will now become a critical component of their cross-functional skill set.

Empowering your team to be successful

Team success is about more than simply being cross-functional, however. Successful small teams must also be empowered to do whatever is necessary to deliver value to their customers. This means each team must be afforded the ability to make the right decisions for their product and their customers in a timely manner.

But this doesn't mean that an agile team must operate outside of authority in order to be successful. The best product managers work within their teams to set a clear vision and strategy that their teams can execute against, as well as build support from the broader organization so they can make those tough decisions when the time comes.

Getting the most from multiplying

Although splitting your organization into multiple smaller teams does have significant advantages over a single large team when it comes to scaling, this approach is not without its challenges.

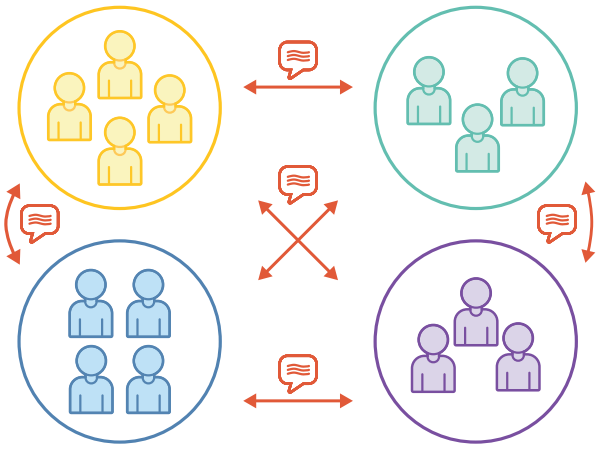
First, the small sizes of these teams means that they are more sensitive to change than larger teams. For example, if a team member is unexpectedly out of the office for a week, this will have a much greater impact on a team of only five people than it would on a team of 15. To help address these types of issues, your team will need to invest heavily in knowledge transfer across the entire team to better prepare them for instances when a key team member is suddenly unavailable.

Second, aligning your teams around specific areas of your product—rather than layers of your product's codebase—will increase each team's ability to deliver the value in their specific area of the product in an independent manner. In addition, reducing the interdependencies between teams by aligning them around specific business problems also reduces the chance of collisions between teams as they work to deliver value to your customers.

Third, effectively sharing the vision for the product only becomes more challenging as the number of teams increases. To combat this, effective product managers take special care to ensure that the vision for the product is properly disseminated across all teams so that the work each team creates is true to that vision. A great way to accomplish this is by holding regular visioning sessions with each team to better communicate to them the long-term vision for the product, as well as where they fit into that vision. Tooling that can provide teams with a high-level view of

the product roadmap—as well as illustrate how their specific tasking contributes to that roadmap—can be invaluable for these discussions.

And finally, you'll need a plan for how each of these individual teams will interact successfully with one another. Increasing the number of separate teams contributing to a product will also increase the complexity of delivering that product. This is because communication pathways between teams increase exponentially as the number of teams grows, just as communication pathways between individuals increase as the size of a team grows. Therefore, you'll need to take special steps to keep all of your teams on the same page so they can work together as effectively as possible.



One way to do this is by synchronizing the iteration schedules of all teams. This means that all teams should operate in iterations of consistent lengths that begin and end on the same days. The result will be a single and aligned schedule that will greatly simplify the long-term planning for your product.

Encouraging your teams to adopt consistent methods and tooling can be a great way to help your teams work together more efficiently.

In addition, encouraging your teams to adopt consistent methods and tooling can be a great way to help your teams work together more efficiently. Having consistent methods and tooling across teams not only removes a common point of friction that can arise when multiple teams need to collaborate to address a shared problem, but it also reduces the challenges that can occur when a team member from one team needs to relocate to another team.

Although long-running teams are ideal, change is inevitable in many organizations. Encouraging a consistent approach to solving problems increases the portability of your team members and allows you to take advantage of opportunities to arrange your teams in the most effective manner.

Keeping a consistent vision

Once the iterations of all of your teams are in alignment and all teams are working in a relatively consistent manner, you can begin kicking off these iterations with collaborative planning sessions involving all teams. Although each team will still need to create their own more detailed plans for their area of work, beginning these planning sessions with a high-level vision of what the next iteration should accomplish can be very valuable for helping all of your teams understand how they must work together to accomplish that vision. A similar event could also be held at the completion of each iteration, in which the work each team completed during that iteration can be shared with other teams. These sessions can be invaluable for helping your teams understand how they are working in concert to realize the product vision.

However, this collaboration must not only happen at the beginning and end of an iteration. For teams to truly work together effectively, they must also stay in touch throughout the entire iteration. Regular check-in and coordination sessions between representatives of each team will ensure that everyone is on the right track to delivering a great product vision, while tooling that emphasizes communication across teams and provides visibility into each team's progress can help ensure that all teams are working together as effectively as possible.

For teams to truly work together effectively, they must also stay in touch throughout the entire iteration.

Making this work for you

Once they've embraced agile practices, few organizations doubt the positive impact it can make on how they deliver software to their customers. However, even after seeing success with their first foray into agile practices, many organizations find that scaling that success beyond a single team is not for the faint of heart.

Here are a few tips that will help your organization find widespread success with scaling its agile practice:

- Prioritize **multiple small teams** over a single large team.
- **Organize** your teams around **distinct feature areas and business functions** rather than layers of your codebase.

- Create teams that are **cross-functional** and possess all of the skills necessary to deliver working software to your customers.

- Encourage **consistent tooling and methods across teams** to enable them to more easily interact and to increase the portability of individual team members.

Scaling for success

Many organizations find success early on when adopting an agile process with a single team. But the real challenge often occurs once that organization begins to build on its success by scaling agile to a larger part of the organization.

While simply increasing the size of that first successful team may often seem like a natural next step, many organizations find that a more promising path to success is to replicate the practices of their first agile teams with other teams in their organization. This more promising path is proof that agile doesn't scale—it multiplies.

About the author

Jeremy Jarrell is an agile coach who helps teams get better at doing what they love. When he's not mentoring Scrum Masters or Product Owners, Jeremy loves to write on all things agile. You can read more of his thoughts at www.jeremyjarrell.com, see his videos at Pluralsight, or follow him on Twitter @jeremyjarrell.